Compress/Decompress Service

Summary

The **e-government framework** adopts Compress of Jakarta Commons providing convenient API to developers.

Support tar, zip and bzip2 files supported at Jakarta Commons.

At present, Commons Compress 1.0-SNAPSHOT API provides

org.apache.commons.compress.archivers
org.apache.commons.compress.archivers.ar
org.apache.commons.compress.archivers.cpio
org.apache.commons.compress.archivers.jar
org.apache.commons.compress.archivers.tar
org.apache.commons.compress.archivers.zip
org.apache.commons.compress.changes
org.apache.commons.compress.compressors
org.apache.commons.compress.compressors.bzip2
org.apache.commons.compress.compressors.gzip
org.apache.commons.compress.utils

Packages.

See Commons Compress 1.0-SNAPSHOT API for more details.

Description

Compression is the technology that compresses the information saved in the file and saves in the smaller memory space.

In general, it reduces size of space required for storage using the method of expressing information or using shorter code or deleting the duplicated contents contained in the information.
Such process is called compression and the process of restoring this to the original status in order to use the compressed information is called decompression.
There are compression method without loss and compression method with loss in the compression methods.
At first, information such as program data should use compression method without loss.
Compression method without loss is the compression method that has the same contents and size as the previous status before compression if restoring the compressed information.
However, compression method with loss can be used in case of image or voice.
It is because there exist massive amounts of image or voice, and the difference cannot be identified with eyes or ears of person even if some of information disappears.
Compression method with loss means the compression method that can have the contents same as the size or the status before compression even if restoring the compressed information.

See the following links for the kinds of compressed files frequently used.

- Kinds of Compressed File

Example

Following consists of compress/decompress using ArchiveInputStream class that belongs to org.apache.commons.compress.archivers package with testZipArchiveCreation() method of test code EgovZipTestCase.

```
public final class EgovZipTestCase extends EgovAbstractTestCase {

        /*
         * Compression(Archive) TEST using Zip
```

```
     * Compressing 2 files(test1.xml, test2.xml) to bla.zip
     */
    public void testZipArchiveCreation() throws Exception {
        // Archive
        final File output = new File(dir, "bla.zip");
        final File file1 = getFile("test1.xml");
        final File file2 = getFile("test2.xml");

        {
            final OutputStream out = new FileOutputStream(output);
            final ArchiveOutputStream os = new
ArchiveStreamFactory().createArchiveOutputStream("zip", out);

            os.putArchiveEntry(new ZipArchiveEntry("testdata/test1.xml"));
            IOUtils.copy(new FileInputStream(file1), os);
            os.closeArchiveEntry();

            os.putArchiveEntry(new ZipArchiveEntry("testdata/test2.xml"));
            IOUtils.copy(new FileInputStream(file2), os);
            os.closeArchiveEntry();
            os.close();
        }

        // Unarchive the same
        List results = new ArrayList();

        {
            final InputStream is = new FileInputStream(output);
            final ArchiveInputStream in = new ArchiveStreamFactory().createArchiveInputStream("zip",
is);

            File result = File.createTempFile("dir-result", "");
            result.delete();
            result.mkdir();

            ZipArchiveEntry entry = null;
            while((entry = (ZipArchiveEntry)in.getNextEntry()) != null) {
                File outfile = new File(result.getCanonicalPath() + "/result/" + entry.getName());
                outfile.getParentFile().mkdirs();
                OutputStream out = new FileOutputStream(outfile);
                IOUtils.copy(in, out);
                out.close();
                results.add(outfile);
            }
            in.close();
        }

        assertEquals(results.size(), 2);
        File result = (File)results.get(0);
        assertEquals(file1.length(), result.length());
        result = (File)results.get(1);
        assertEquals(file2.length(), result.length());
    }
```

Following is the test code decomopressing (Unarchive) the compressed file above.

```
    public void testZipUnarchive() throws Exception {
        final File input = getFile("bla.zip");
        final InputStream is = new FileInputStream(input);
        final ArchiveInputStream in = new ArchiveStreamFactory().createArchiveInputStream("zip",
is);
        final ZipArchiveEntry entry = (ZipArchiveEntry)in.getNextEntry();
        final OutputStream out = new FileOutputStream(new File(dir, entry.getName()));
```

```
        IOUtils.copy(in, out);
        out.close();
        in.close();
    }
```

N. Reference

Compression Algorithm

Apache Commons Compress

Commons Compress 1.0-SNAPSHOT API